



## Inversión de control

### Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de Servlet. En el video pasado vimos que web.xml tiene una ventaja en relación a nuestras anotaciones, que es que todas las configuraciones quedan en un único lugar, mientras que si nosotros usamos por ejemplo anotaciones, tenemos que colocarla en varios lugares, en varios archivos.

[00:24] Pero tiene una ventaja muy grande, usar anotaciones, que es que simplifica mucho el código realmente. Entonces, ahora en esta vez me gustaría hablar un poco más sobre nuestro Servlet y Tomcat. Me gustaría que vean lo siguiente. Nuestro Servlet es un objeto, un objeto que nosotros creamos bien y nuestro Tomcat nos permite que a través de una petición Http, nosotros podamos acceder a ese objeto.

[00:55] Entonces eso es una pregunta de certificaciones. Si ustedes van a hacer alguna certificación sobre Servlet, Tomcat, van a ver que está esa pregunta. ¿Qué es un Servlet? Es un objeto que podemos acceder a él a través de una petición HTTP por medio de nuestro navegador y que nuestro Tomcat quién va a hacer ese gerenciamento de pasaje de información.

[01:20] Entonces, si es un objeto significa que alguien lo creó. Entonces vamos a ir a nuestro código y vean lo siguiente. En ningún lugar, en ningún lugar de nuestro código nosotros hemos hecho un main. En ningún lugar hicimos esto. En ningún lugar hicimos un `HolaMundoServlet hola = new HolaMundoServlet();` en ningún lugar hicimos eso y en ningún lugar hicimos `hola.service` ni le pasamos los objetos como parámetro.

[01:54] Y sin embargo, ustedes saben que en todo programa con Java, siempre el main es el método por el cual se inicia una aplicación, entonces si no lo hemos hecho nosotros ¿quién lo ha hecho? Bueno, quien lo hizo fue realmente nuestro Tomcat. Nuestro Tomcat es nuestro Servlet container. Un servlet container es un contenedor de Servlets.

[02:20] Entonces vean la importancia de nuestro de nuestro Tomcat. Él hace que, por ejemplo, se creen todos esos Servlets. Nuestro Tomcat se dice también que es un intermediario. Si vemos acá en el gráfico, Tomcat es lo que está entre nuestros objetos que nosotros creamos y las peticiones http que nos envía nuestro navegador.

[02:43] Entonces Tomcat se dice también que es un middleware, alguien que está en el medio haciendo ese gerenciamiento, pasaje de información de un lado al otro y para probarlo voy a mostrarles acá. Vamos a colocar un constructor HolaMundoServlet. Saben que toda clase tiene un constructor, aun cuando nosotros no colocamos ninguno.

[03:06] Siempre existe un constructor padrón, bien predefinido, entonces en este caso vamos a utilizar nuestro propio constructor y voy a hacer un `System.out.println`. Y acá voy a colocar "Creando Hola Mundo Servlet". "Ctrl + S". Voy a iniciar de nuevo nuestro Tomcat. Y vean lo siguiente.

[03:31] En la consola, cuando yo llamo a nuestro hola, vean que se creó nuestro Creando Hola Mundo Servlet y un se ejecutó el Servlet, que es un código que tenemos acá dentro de nuestro service. Pero vea lo siguiente, si yo inicio de nuevo nuestro servidor, voy a dar un run, vean que en ningún lugar está ese creando Hola Mundo Server. ¿Qué significa eso?

[04:00] Significa que nuestra clase aún no fue instanciada, o sea, no se creó una instancia de nuestra clase todavía, aun cuando hemos levantado nuestro servidor, o sea que cuando levantamos el servidor no tenemos una instancia por cada Servlet. Esos Servlets son creados a medida que son necesarios,

entonces por ejemplo acá, cuando yo voy a dar un F5, ahora sí se creó nuestro Servlet y se ejecutó nuestro Servlet.

[04:28] Eso significa que nuestro Tomcat es perezoso, o sea, no va a crear una instancia de una clase si no es necesaria aún. Y no solo eso, es bastante perezoso. Si yo hoy muchos F5 acá, vean en la consola que una sola vez fue ejecutada, se creó esa instancia de nuestro Servlet y se ejecutó muchas veces.

[04:53] Pero solamente existe una instancia de nuestro Servlet. ¿Qué significa? Que en nuestra memoria, o sea, una vez que nosotros creamos nuestro Servlet, queda en memoria. Y queda ahí durante el tiempo que viva nuestro servidor. Si el servidor está levantado y nosotros llamamos una vez, aunque sea ese Servlet, entonces está en memoria.

[05:16] Entonces, si solamente crea una instancia de cada Servlet, significa que ese Servlet es un singleton. Singleton significa que existe una sola instancia. ¿Cuánto tiempo vive un singleton? Singleton significa para siempre, o sea, que vive para siempre, mientras esté levantado nuestro servidor.

[05:40] Cada singleton que se crea se dice que tiene un scope de singleton. El scope o la vida es durante toda la vida del server. Tomcat podría, por ejemplo, crear una instancia de cada Servlet cada vez que se envía una request, cada vez que nuestro navegador envía una petición, un HTTP con una request, nuestro Tomcat podría si quisiera crear una instancia de cada Servlet cada vez.

[06:11] Pero no es así como funciona nuestro Tomcat. Existen otros frameworks que sí hacen eso, que cada vez que se envía una request se crea una instancia nueva, pero nuestro Tomcat no hace eso. ¿Por qué les hablo tanto sobre este tema? Es porque este tema está dentro de otro tema llamado inversión de control.

[06:31] ¿Qué es la inversión de control? Es dejar que otra cosa, en este caso Tomcat, haga cosas en vez de nosotros. En este caso, por ejemplo, yo no creo nuestro método main, yo le dejo a Tomcat que Tomcat cree el método main.

Entonces le estoy cediendo el control de esa tarea a Tomcat. ¿Por qué le estoy cediendo esa tarea a Tomcat? Porque él sabe hacer bien, manejar bien las peticiones HTTP.

[07:01] Si yo quisiera, por ejemplo, que en cada proyecto que yo hago yo mismo crear toda la lógica alrededor de poder recibir peticiones HTTP, enviar y manejar todo ese protocolo, realmente vean que sería muy parecido todo eso que yo creería que sería muy parecido dentro de cada proyecto.

[07:19] Entonces yo directamente le dejé a Tomcat que se encargue de eso y yo me centro en la lógica, Tomcat ya sabe cómo trabajar con ese HTTP. Entonces nos da todos esos datos que viene en HTTP nos lo devuelve acá en forma de objetos ya formateados y ya lindos y listos para nosotros ya utilizar. Tomcat podría hacer más cosas, por ejemplo, si quisiera.

[07:40] Tomcat podría por ejemplo gerenciar lo que es la parte de hacer un análisis de seguridad de los datos que nos envían o Tomcat podría por ejemplo gerenciar las transacciones con la base de datos, o podría gerenciar la parte de catch, o caché como le decimos en español.

[08:03] Pero Tomcat no hace eso. Toda esa parte queda para algunos frameworks más sofisticados, como por ejemplo Spring. Spring. Es uno de los frameworks más sofisticados dentro del mundo. Entonces, con todo esto que les he dicho, me gustaría que se queden con toda esa parte de singleton y el significado de la inversión de control, que es yo dejar que esta herramienta, que Tomcat en este caso haga cosas en vez de yo tener que hacerlas manualmente.

[08:30] Y además hace algunos extras en el caso de algunos frameworks, como por ejemplo todo ese gerenciamiento de transacciones de base de datos, catch, análisis de seguridad, etcétera. Dicho todo esto, vamos a hacer los ejercicios y nos vemos en el próximo video.

